Chebyshev lattices as unifying framework for point sets used in interpolation and integration

Koen Poppe - Ronald Cools

Overview

1. Generalisation to s-D
2. Chebyshev lattice rules
3. Classification existing rules
4. Search
5. Results
6. Conclusion

1. Generalisation to s-D

1 dimensional: $f(x), x \in [-1, 1]$

- Chebyshev series $f(x) = \sum_{h=0}^{\infty} \alpha_h \, \hat{T}_h(x)$

- Interpolating approx. $f(x) \approx \sum_{h=0}^{n} a_h \, \hat{T}_h(x)$

- Quadrature rule (Clenshaw-Curtis) $\int_{-1}^{1} f(x) \, \mathrm{d}x \approx \sum_{h=0}^{n} a_h \, I_{[\hat{T}_h]}$

1. Generalisation to s-D

s-dimensional: $f(\vec{x}), \vec{x} \in [-1,1]^s$

- Chebyshev series $f(\vec{x}) = \sum_{\vec{h}} \alpha_{\vec{h}} \hat{T}_{\vec{h}}(\vec{x})$

- Hyperinterpolation $f(\vec{x}) \approx \sum_{\vec{h},|\vec{h}|\leq n} a_{\vec{h}} \hat{T}_{\vec{h}}(\vec{x})$

- Cubature rule $\int_{[-1,1]^s} f(\vec{x})\,\mathrm{d}\vec{x} \approx \sum_{\vec{h},|\vec{h}|\leq n} a_{\vec{h}} I_{[\hat{T}_{\vec{h}}]}$

1. Generalisation to s-D

HYPERINTERPOLATION

$$f(\vec{x}) \approx \sum_{\vec{h},|\vec{h}|\leq n} a_{\vec{h}}\, \hat{T}_{\vec{h}}(\vec{x})$$

Projection into polynomial function space

$$a_{\vec{h}} \approx \int_{[-1,1]^s} f(\vec{x})\, \hat{T}_{\vec{h}}(\vec{x})\, \omega(\vec{x})\, \mathrm{d}\vec{x}$$

using cubature rule for weighted setting

$$a_{\vec{h}} = \sum_{l} w_l\, f(\vec{x}_l)\, \hat{T}_{\vec{h}}(\vec{x}_l)$$

1. Generalisation to s-D

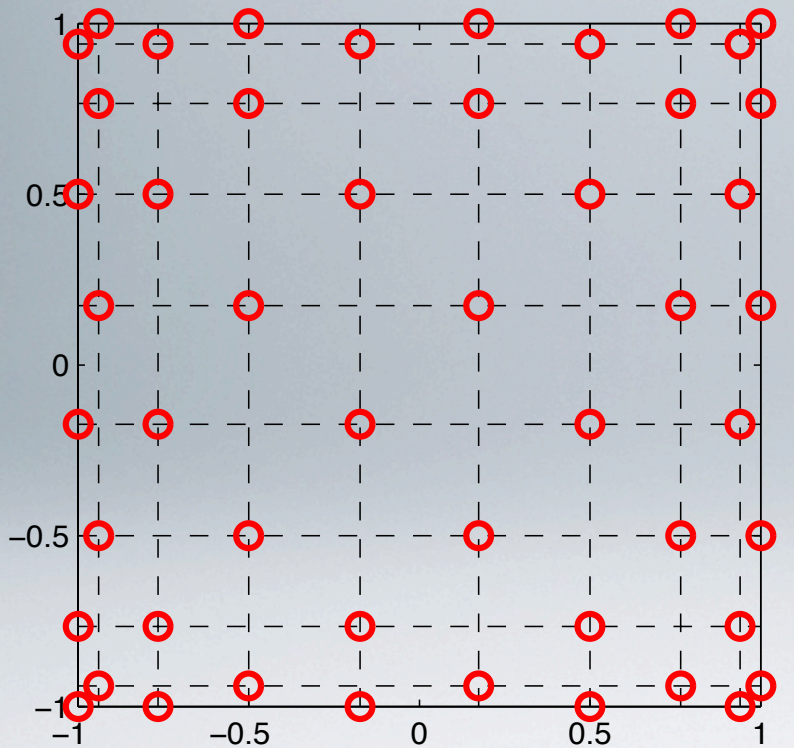
WEIGHTED CUBATURE

$$Q[g] = \sum_{l} w_l \, g(\vec{x}_l) \approx \int_{[-1,1]^s} g(\vec{x}) \, \omega(\vec{x}) \, \mathrm{d}\vec{x}$$

- Several *good* point sets exist
- Known lower bounds on # points
- No clear constructive approach
- Extrema of Chebyshev polynomials

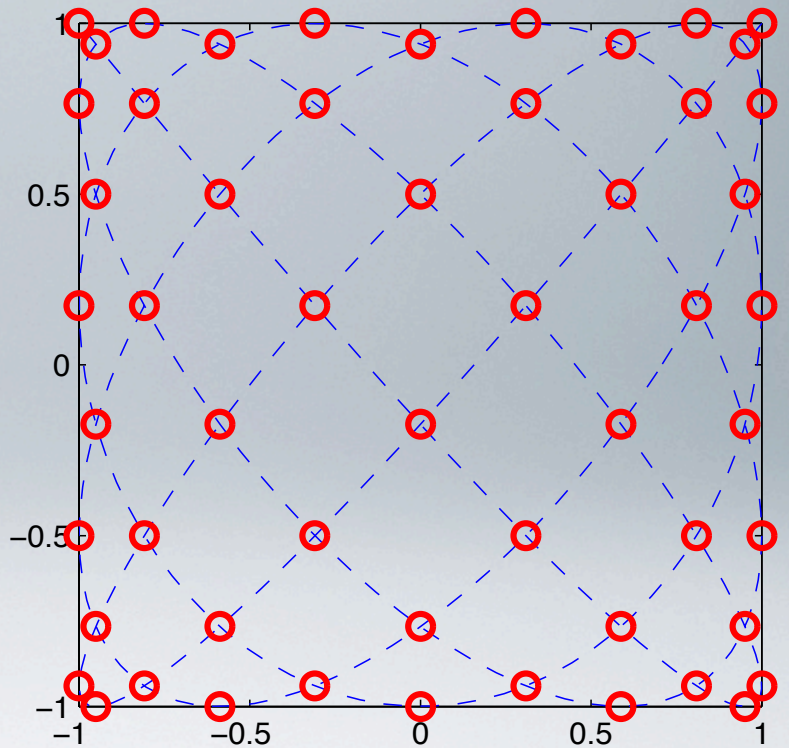
1. Generalisation to s-d

POINT SET EXAMPLES



Morrow-Patterson

(°1978, deg. 17, #50)



Padua

(°2005, deg. 17, #55)

Overview

1. Generalisation to s-D
2. Chebyshev lattice rules
3. Classification existing rules
4. Search
5. Results
6. Conclusion

2. Chebyshev lattice rules

OBJECTIVES

1. Increase number of component values
2. Easy extendible to higher dimensions
3. Compact and uniform representation
4. Enable the use of the FFT
5. Provide a constructive approach

2. Chebyshev lattice rules

CHEBYSHEV LATTICE

$$\vec{x}_{\vec{\ell}} = \cos\left(\pi\left(\frac{\ell_1\,\vec{z}_1}{d_1} + \cdots + \frac{\ell_k\,\vec{z}_k}{d_k} + \frac{\vec{z}_\Delta}{d_\Delta}\right)\right)$$

$$\vec{z}_1, \ldots, \vec{z}_k, \vec{z}_\Delta \in \mathbb{Z}^s \qquad d_1, \ldots, d_k, d_\Delta, \ell_1, \ldots, \ell_k \in \mathbb{Z}$$

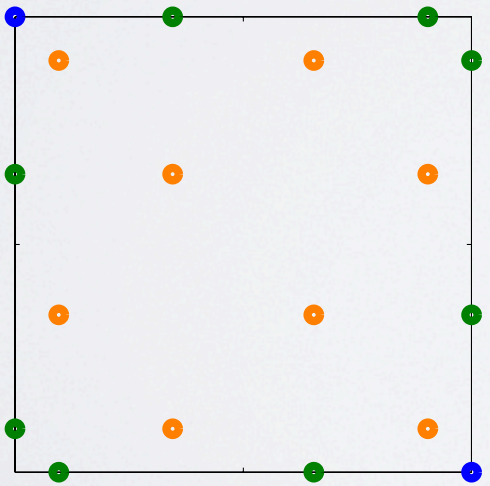
- rank-k integration lattice
- offset vector
- mimic Chebyshev extrema

2. Chebyshev lattice rules

CHEBYSHEV LATTICE RULE

$$\vec{x}_{\vec{\ell}} = \cos\left(\pi\left(\frac{\ell_1\,\vec{z}_1}{d_1} + \cdots + \frac{\ell_k\,\vec{z}_k}{d_k} + \frac{\vec{z}_\Delta}{d_\Delta}\right)\right)$$

$$w_{\vec{\ell}} = \text{cte}\left(\frac{1}{2}\right)^{\#_r(|x_{\vec{\ell},r}|=1)}$$

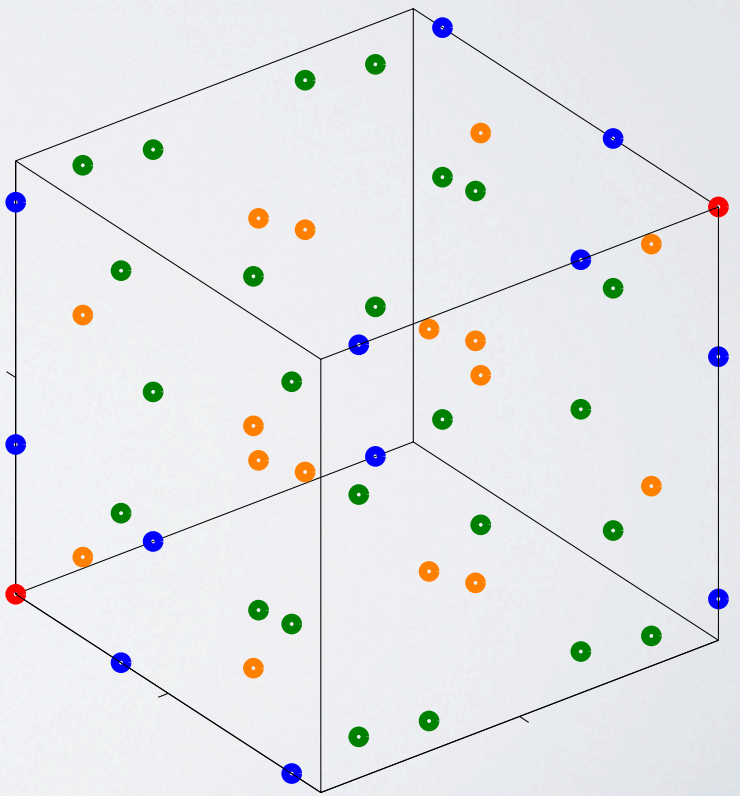


cte

cte/2

cte/4

cte/8



Overview

1. Generalisation to s-D
2. Chebyshev lattice rules
3. Classification existing rules
4. Search
5. Results
6. Conclusion

3. Classification existing rules

Clenshaw-Curtis (1960)

Morrow-Patterson (1978)

n=11, fully symmetric (1988)

Minimal D4-invariant (1989)

Blending, Godzina (1994)

Padua (2005)

Xu (2009)

Padua-like (-)

3. Classification existing rules

	k=s (full rank)	k=s-1
s=1	Clenshaw-Curtis (1960)	—
s=2	Morrow-Patterson (1978) n=11, fully symmetric (1988) Minimal D4-invariant (1989)	Padua (2005)
s=3	Blending, Godzina (1994) *Xu (2009)*	*Padua-like (-)*
s	Blending, Godzina (1994)	*Padua-like (-)*

Overview

1. Generalisation to s-D
2. Chebyshev lattice rules
3. Classification existing rules
4. Search
5. Results
6. Conclusion

4. Search

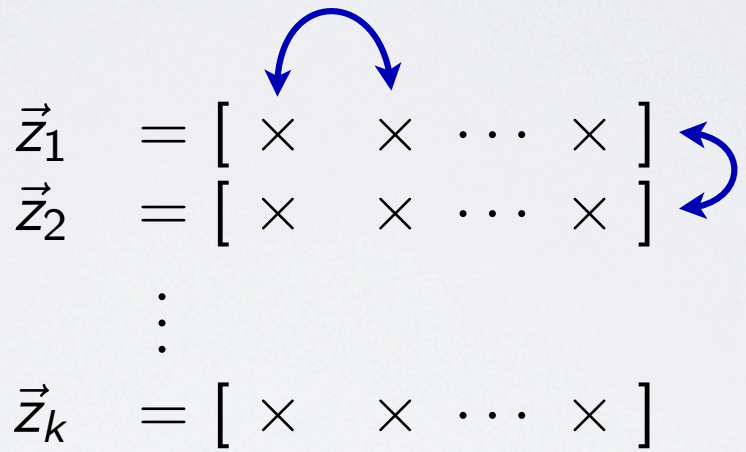
SETUP

- In 2, 3, 4 and 5-D, not limited to rank-1
- Most basic search algorithm:
  - ➡ for each d
    - ➡ for each $\{\vec{z}_1, \ldots, \vec{z}_k\}$
      - create the point set
      - verify its degree
      - store rule if better
- Rank and dimension as parameters

4. Search

SEARCH SPACE REDUCTION

- Size search space: $(d+1)^{ks}$
- Optimisation: exploit symmetries



- How to generate these vectors?

4. Search

SEARCH SPACE GENERATION

- Parametric representation
  - i.e. s=3, rank-1:

$$[\mathtt{A},\mathtt{A},\mathtt{A}]$$
$$[\mathtt{A},\mathtt{A},\mathtt{B}]$$
$$[\mathtt{A},\mathtt{B},\mathtt{B}]$$
$$[\mathtt{A},\mathtt{B},\mathtt{C}]$$

for $\mathtt{A},\mathtt{B},\mathtt{C} = 0,\ldots,d$
$\mathtt{A} < \mathtt{B} < \mathtt{C}$

  - i.e. s=2, rank-2:

$$[\mathtt{A},\mathtt{A}]\ [\mathtt{A},\mathtt{B}]$$
$$[\mathtt{A},\mathtt{B}]\ [\mathtt{A},\mathtt{C}]$$
$$[\mathtt{A},\mathtt{B}]\ [\mathtt{B},\mathtt{A}]$$
$$[\mathtt{A},\mathtt{B}]\ [\mathtt{C},\mathtt{A}]$$
$$\ldots$$

for $\mathtt{A},\mathtt{B},\mathtt{C},\mathtt{D} = 0,\ldots,d$
$\mathtt{A} < \mathtt{B} < \mathtt{C} < \mathtt{D}$

- Parameters through (almost) loopless algorithm

4. Search

POINT SET CREATION

- Problem: duplicates
- Red-black tree
  comparison is bottleneck ...
    - avoid floating point
      - reduce integer parts to $[0, d]$: $\vec{y}_\ell = \text{minmod}\,(\ell\vec{z})$
    - exploit large integer length
      - composite scalar: $y_\ell = \sum_{r=1}^{s} (d+1)^{r-1}\,\text{minmod}\,(\ell z_r)$

$$\vec{x}_\ell = \cos\left(\frac{\pi}{d}\,\ell\vec{z}\right)$$

integer

4. Search

DEGREE VERIFICATION

- Test inner cubature rule for all $\hat{T}_{\vec{h}} \quad (|\vec{h}| \leq n)$ evaluation is bottleneck
  - cache intermediate results
    - trade off: storage vs. computations
  - change order of coefficients $\vec{h}$
    - maximise intermediate cache “hits”
    - (almost) loopless algorithm

Overview

1. Generalisation to s-D
2. Chebyshev lattice rules
3. Classification existing rules
4. Search
5. Results
6. Conclusion

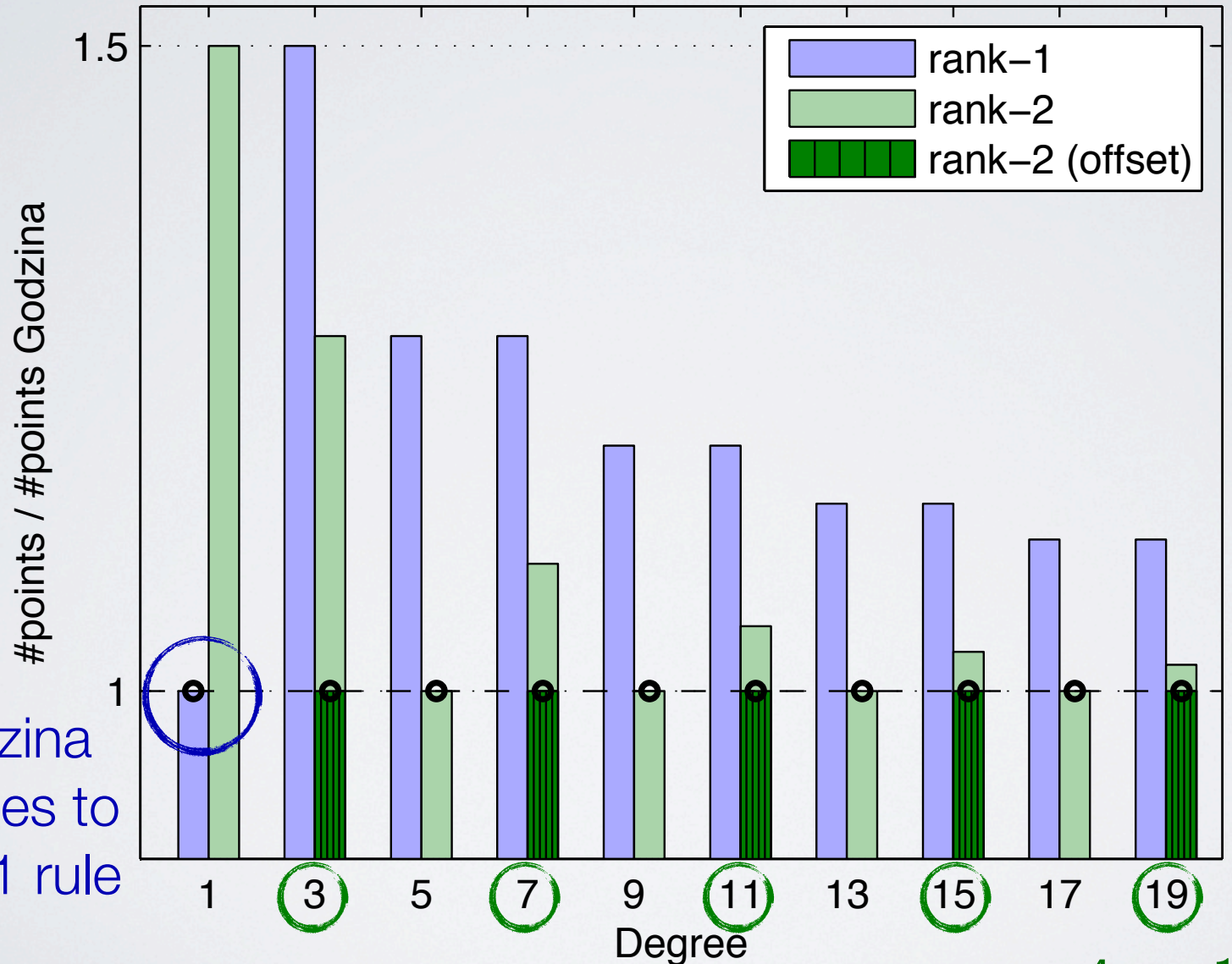
5. Results

GENERAL REMARKS

- Distinct component values still small
  - grid-like structure decreases #points
- No point sets better than already known before
  - Godzina best
- Observations:
  - high rank consistently better than low rank
  - offset vectors often reduce # points

5. Results

2-DIMENSIONS

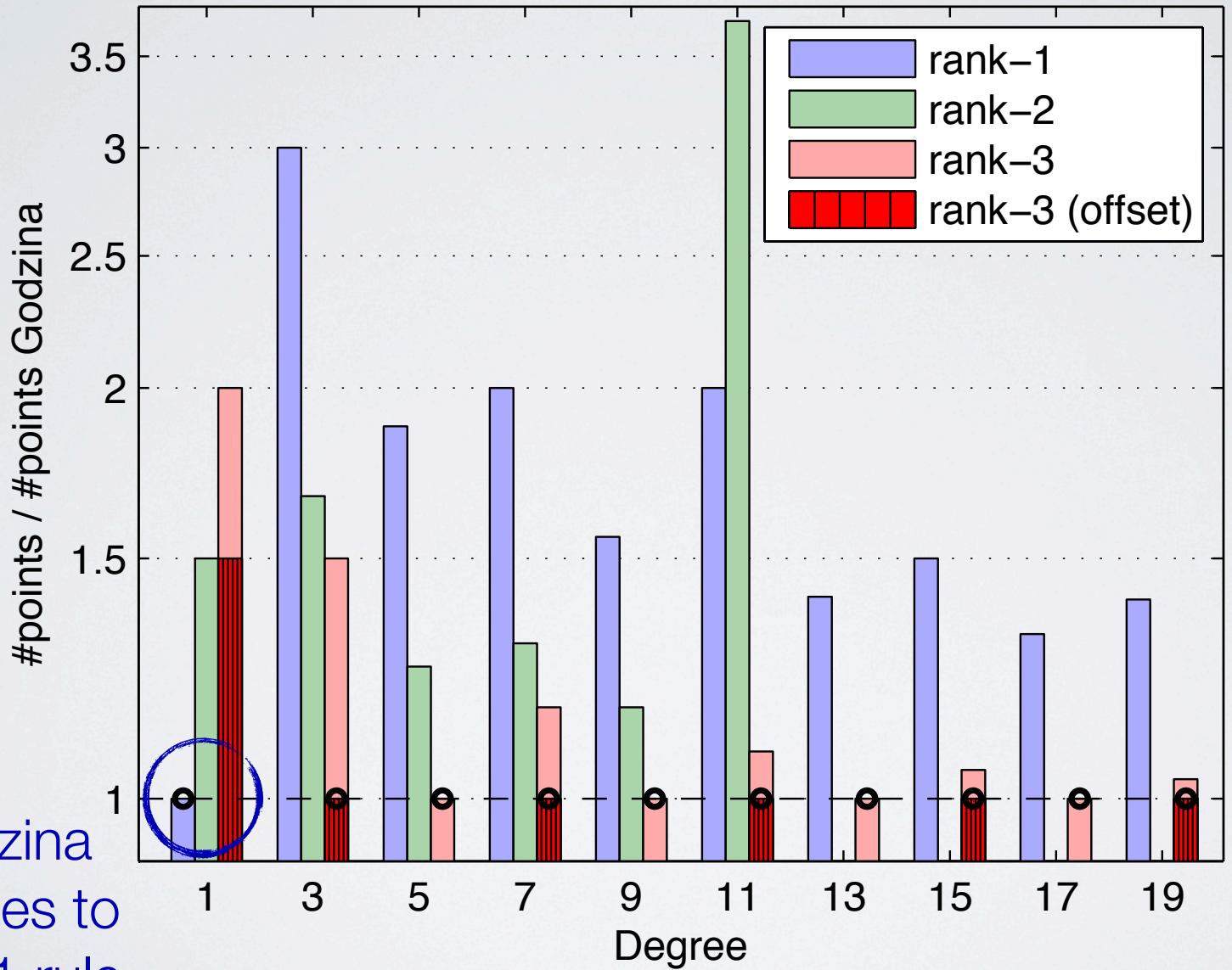


Godzina reduces to rank-1 rule

$n = 4\nu - 1$ with offset

5. Results

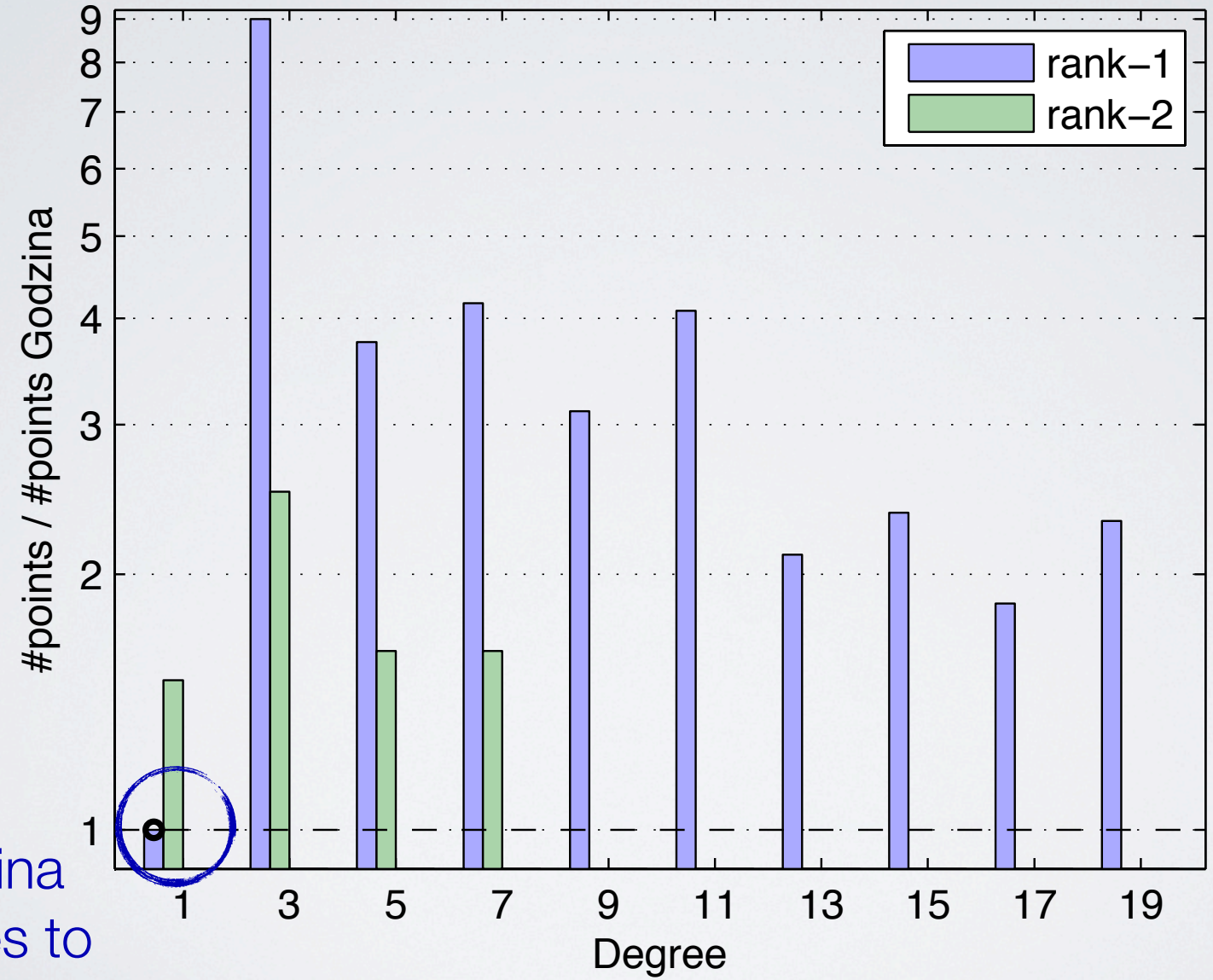
3-DIMENSIONS



Godzina reduces to rank-1 rule

5. Results

4-DIMENSIONS (INCOMPLETE)



Godzina reduces to rank-1 rule

6. Conclusion

- Multivariate extension of Clenshaw-Curtis
- Chebyshev lattices as general framework
- Uniform description for known point sets
- High rank computational search
  - Godzina blending formulae still best
  - Rank-1 rules interesting special case

Future work

- Cubature rules suited for oscillatory functions